
NuMI Online Beamline Monitoring using JAS *and Offline Data Analysis Results*

Mary Bishai, Mark Dierckxsens, Brett Viren

`mbishai@bnl.gov, bviren@bnl.gov, mdier@bnl.gov`

Brookhaven National lab

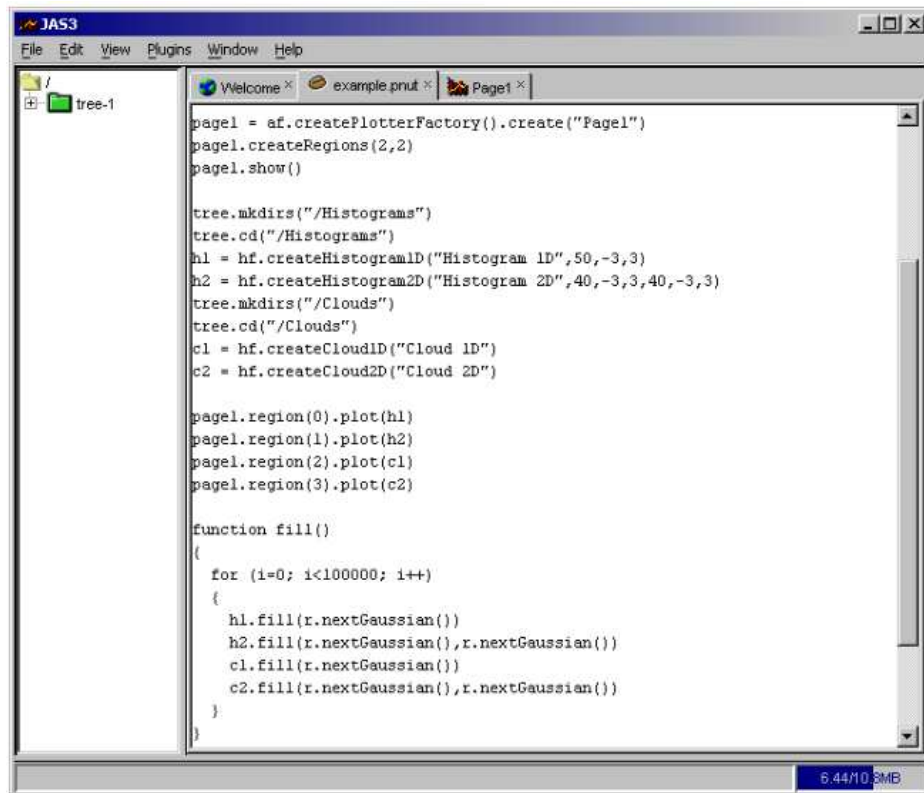
What Is JAS? Unnecessary Jargon:

The goal of the **Abstract Interfaces for Data Analysis (AIDA)** project is to “define abstract interfaces for common physics analysis objects, such as histograms, ntuples, fitters, IO etc.. The adoption of these interfaces should make it easier for physicists to use different tools without having to learn new interfaces or change all of their code. Additional benefits will be interoperability of AIDA compliant applications (for example by making it possible for applications to exchange analysis objects via XML).”

<http://java.freehep.org/jaida/index.html>

- **JAIDA** is the **JAVA** language implementation of **AIDA**.
- **Java Analysis Studio 3 (JAS3)** is a full featured GUI which uses JAIDA internally and runs in real time.

JAS Examples - Histogramming



```
File Edit View Plugins Window Help
Welcome x example.pnut x Page1 x
tree-1

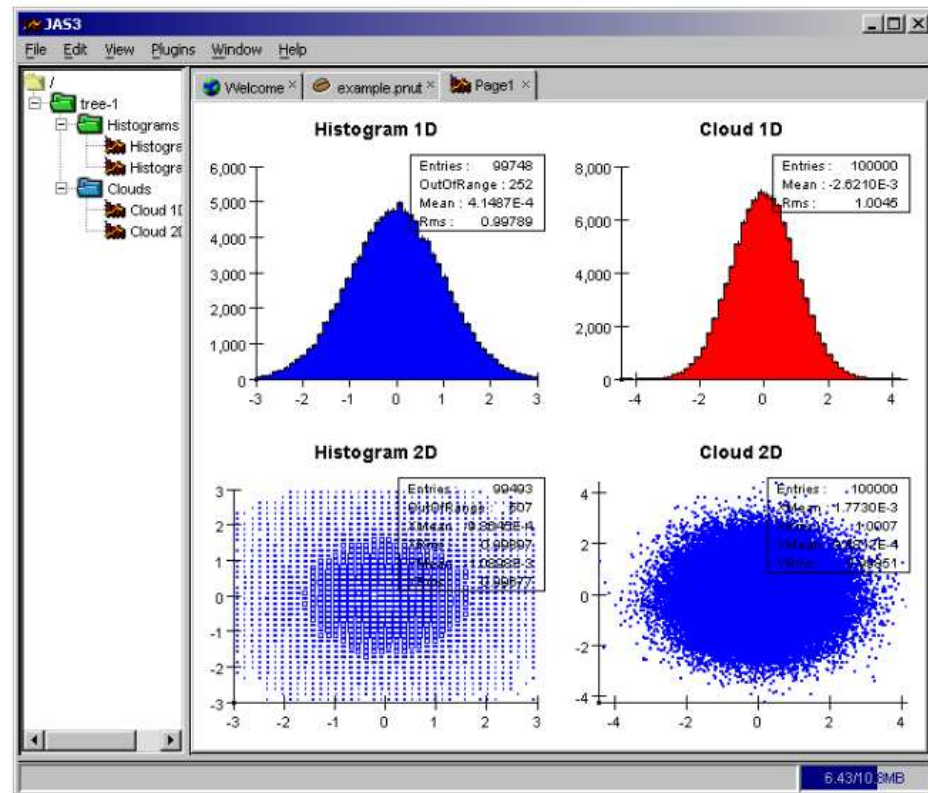
page1 = af.createPlotterFactory().create("Page1")
page1.createRegions(2,2)
page1.show()

tree.mkdir("/Histograms")
tree.cd("/Histograms")
h1 = hf.createHistogram1D("Histogram 1D",50,-3,3)
h2 = hf.createHistogram2D("Histogram 2D",40,-3,3,40,-3,3)
tree.mkdir("/Clouds")
tree.cd("/Clouds")
c1 = hf.createCloud1D("Cloud 1D")
c2 = hf.createCloud2D("Cloud 2D")

page1.region(0).plot(h1)
page1.region(1).plot(h2)
page1.region(2).plot(c1)
page1.region(3).plot(c2)

function fill()
{
  for (i=0; i<100000; i++)
  {
    h1.fill(r.nextGaussian())
    h2.fill(r.nextGaussian(),r.nextGaussian())
    c1.fill(r.nextGaussian())
    c2.fill(r.nextGaussian(),r.nextGaussian())
  }
}
```

6.44/10 MB



JAS Examples - Fitting

```
JAS3
File Edit View Plugins Window Help

tree-1
  Histogram 11
    gauss

hf = af.createHistogramFactory(tree)
hl = hf.createHistogramID("Histogram 1D",50,-3,3)

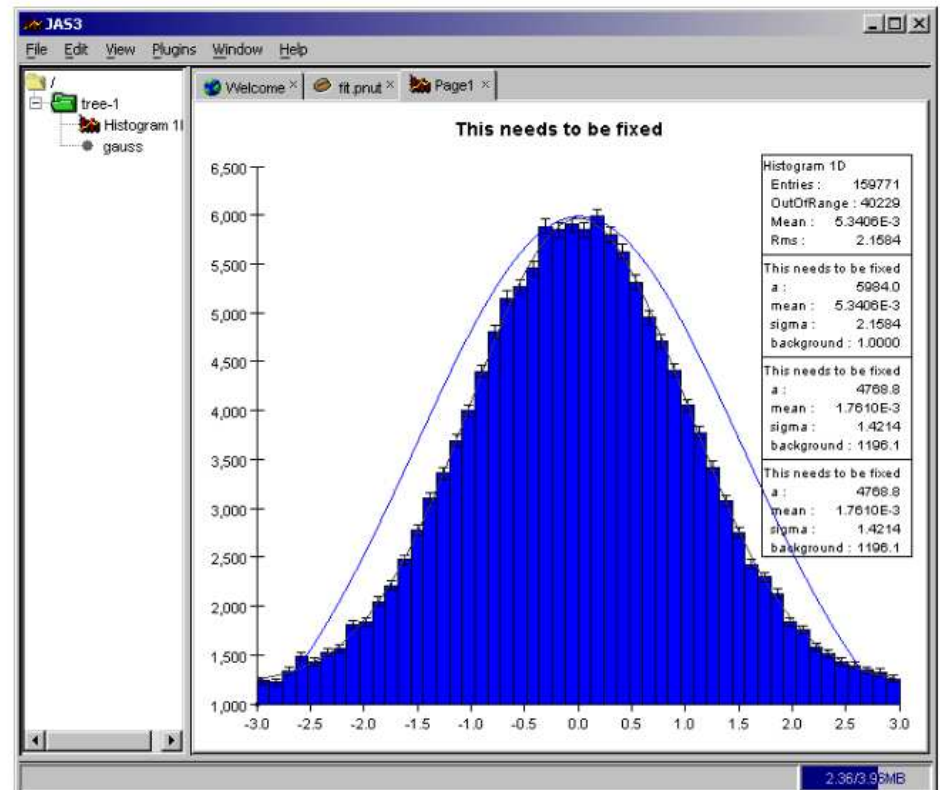
Random = class java.util.Random
r = Random()
for (i=0; i<100000; i++) hl.fill(r.nextGaussian())
for (i=0; i<100000; i++) hl.fill(r.nextDouble()*10-5)

functionfact = af.createFunctionFactory(tree)
gauss = functionfact.createFunctionFromScript("gauss",1,"background+a*exp(-(
gauss.setParameter("a",hl.maxBinHeight())
gauss.setParameter("mean",hl.mean())
gauss.setParameter("sigma",hl.rms())

pagel = af.createPlotterFactory().create("Page1")
pagel.region(0).plot(hl)
pagel.region(0).plot(gauss)

ff = af.createFitFactory()
minuit = ff.createFitter("Chi2","minuit")
uncmin = ff.createFitter("Chi2","uncmin")
minuitResult = minuit.fit(hl,gauss)
uncminResult = uncmin.fit(hl,gauss)
pagel.region(0).plot(minuitResult.fittedFunction())
pagel.region(0).plot(uncminResult.fittedFunction())
pagel.show()

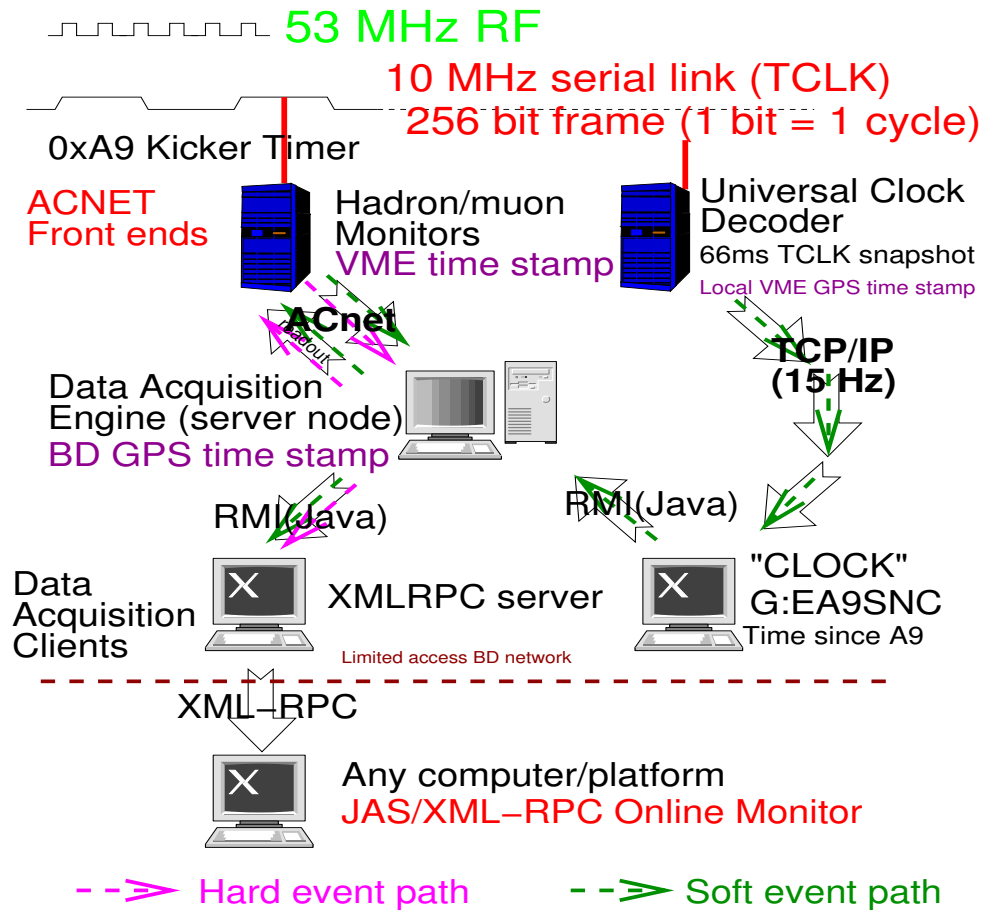
2.46/3.96MB
```



Why Use JAS for NuMI Monitoring?

- Java GUI capabilities well established and **easy to use**.
- Java XML-RPC implementation = **access to ACNET data**.
- JAIDA allows handling of complex data, full HEP data analysis capabilities including **histogramming, fitting and ntupling**.
- JAS provides nice **GUI interface** to JAIDA for **run-time histogram and fitting displays**.
- JAS3 architecture is based on FreeHEP Application Framework into which many optional modules can be plugged in. **Highly customizable**.
- JAS/JAVA already in use by FNAL AD controls (Timofei Bolshakov, Jim Patrick and others) so **local expertise available**.
- Real reason: Too lazy, already knew JAVA, and hate ROOT/C++(M. Bishai).

Getting NuMI Data → JAS Monitor



- A request is sent via HTTP to Charlie Kings' XML-RPC service to extract the data from a list of ACNET devices 1 second after an A9 accelerator event (NuMI kicker fires) occurs.
- The "clock" process listens to a 15 Hz broadcast of TCLK snapshots. "clock" informs the DAE when the snapshot contains the A9 event.
- The DAE requests the data from the ACNET front ends and sends it to the XML-RPC server which broadcasts it to the world

Any computer running Java can monitor ACNET data using

JAS/XML-RPC: <http://minos.phy.bnl.gov/~bishai/minos/NuMIMon/>

LIVE DEMO OF NuMI JAS/XMLRPC MONITOR (time and
network permitting)
If not - show screen snapshots

NuMIMon Screen Shot 1

The XML-RPC interface dialog box:

The image shows two windows from the NuMIMon application. The left window is the 'XML-RPC interface dialog box' titled 'BROOKHAVEN NATIONAL LAB'. It contains several input fields for configuration: 'Server URL' (http://www-bd.fnal.gov:80/xmlrpc-test/export), 'Call back URL' (http://130.199.22.131), 'Call back port #' (19871), 'Trigger cycle name' (A9), 'Trigger cycle delay' (130), 'Device list File' (device.dat), and 'Job list File' (jobs.cfg). Below these fields is a 'Job id list (select job to stop)' dropdown menu set to 'ALL'. There is a checkbox for 'Write data to file. The file name is the job id.' which is currently unchecked. At the bottom are four buttons: 'START JOBS' (green), 'START JOB' (green), 'STOP' (red), and 'EXIT' (grey). The right window is a file list window showing a list of files in a directory, including G:EA9SNC, E:TRTGT, E:HMGPD, E:HMRTD, E:HADMDS, E:MM1GPD, E:MM1RTD, E:MMA1DS, E:MM2GPD, E:MM2RTD, E:MMA2DS, E:MM3GPD, E:MM3RTD, E:MMA3DS, E:M101DS, E:M105DS, E:M107DS, E:M108DS, E:M112DS, E:M114DS, E:M115DS, E:M117DS, E:M121DS, and E:MTGTD. The status bar at the bottom of the right window shows 'device.dat 4:17PM 0.30 Mail (Text CVS:1.5 Fill)--L25--All' and '(No changes need to be saved)'.

BROOKHAVEN NATIONAL LAB

Server URL:

Call back URL:

Call back port #:

Trigger cycle name:

Trigger cycle delay:

Device list File:

Job list File:

Job id list (select job to stop) **ALL** ▼

☐ Write data to file. The file name is the job id.

START JOBS **START JOB** **STOP** **EXIT**

File Edit Options Buffers Tools Help

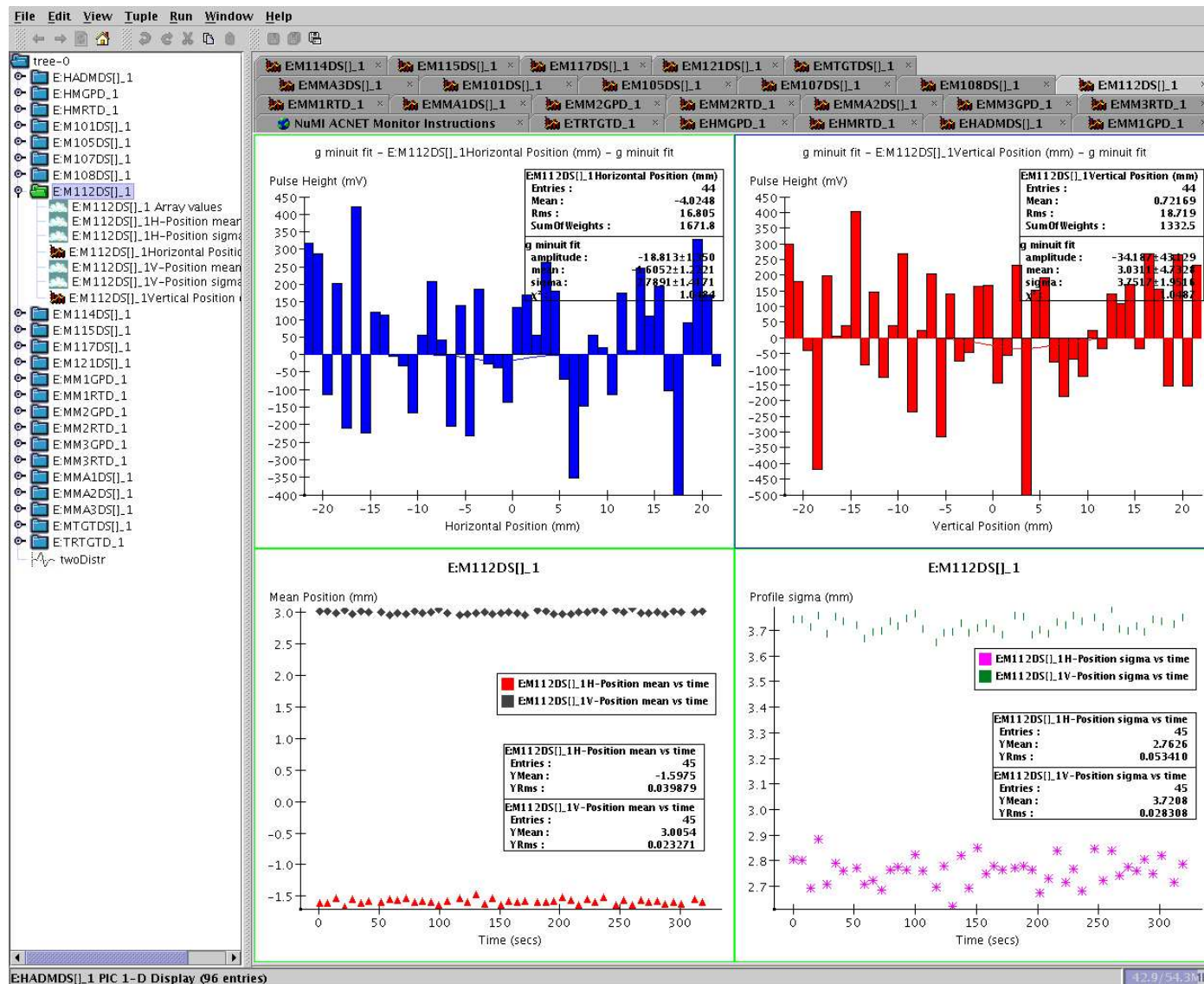
G:EA9SNC
E:TRTGT
E:HMGPD
E:HMRTD
E:HADMDS
E:MM1GPD
E:MM1RTD
E:MMA1DS
E:MM2GPD
E:MM2RTD
E:MMA2DS
E:MM3GPD
E:MM3RTD
E:MMA3DS
E:M101DS
E:M105DS
E:M107DS
E:M108DS
E:M112DS
E:M114DS
E:M115DS
E:M117DS
E:M121DS
E:MTGTD

device.dat 4:17PM 0.30 Mail (Text CVS:1.5 Fill)--L25--All
(No changes need to be saved)

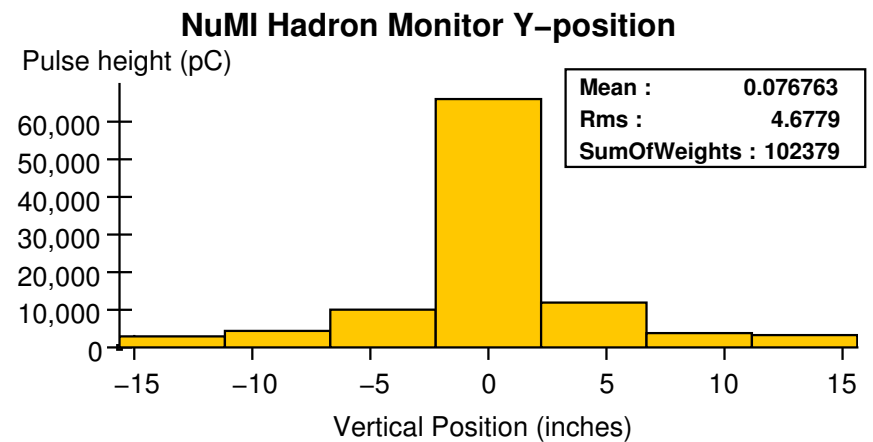
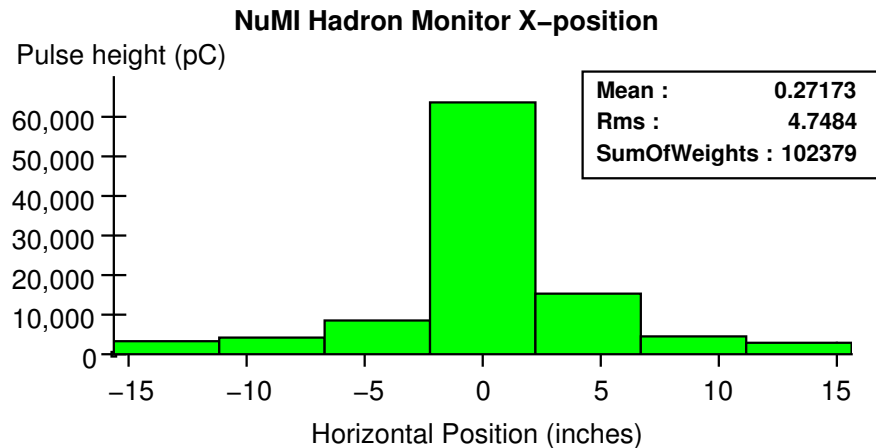
NuMIMon Screen Shot 2



NuMIMon Screen Shot 3

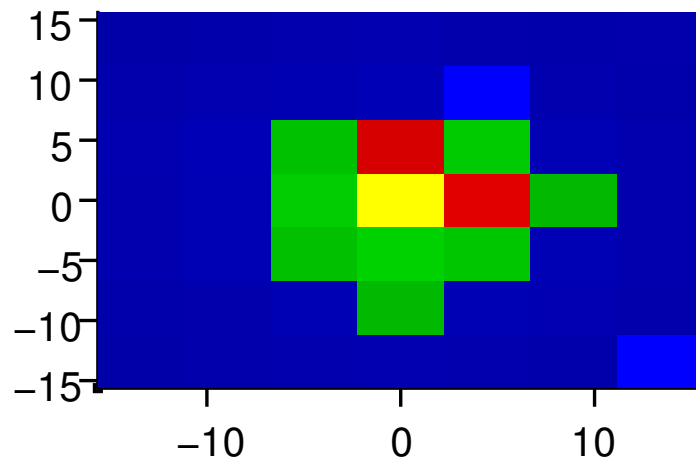


NuMI First Beam !



NuMI Hadron Monitor 2-D Display (log Z)

Vertical position (inches)



XMean : 0.27173
XRms : 4.7484
YMean : 0.076763
YRms : 4.6779
SumOfWeights : 102379

OFFLINE ANALYSIS OF NUMI DATA FROM DEC 3-4th.

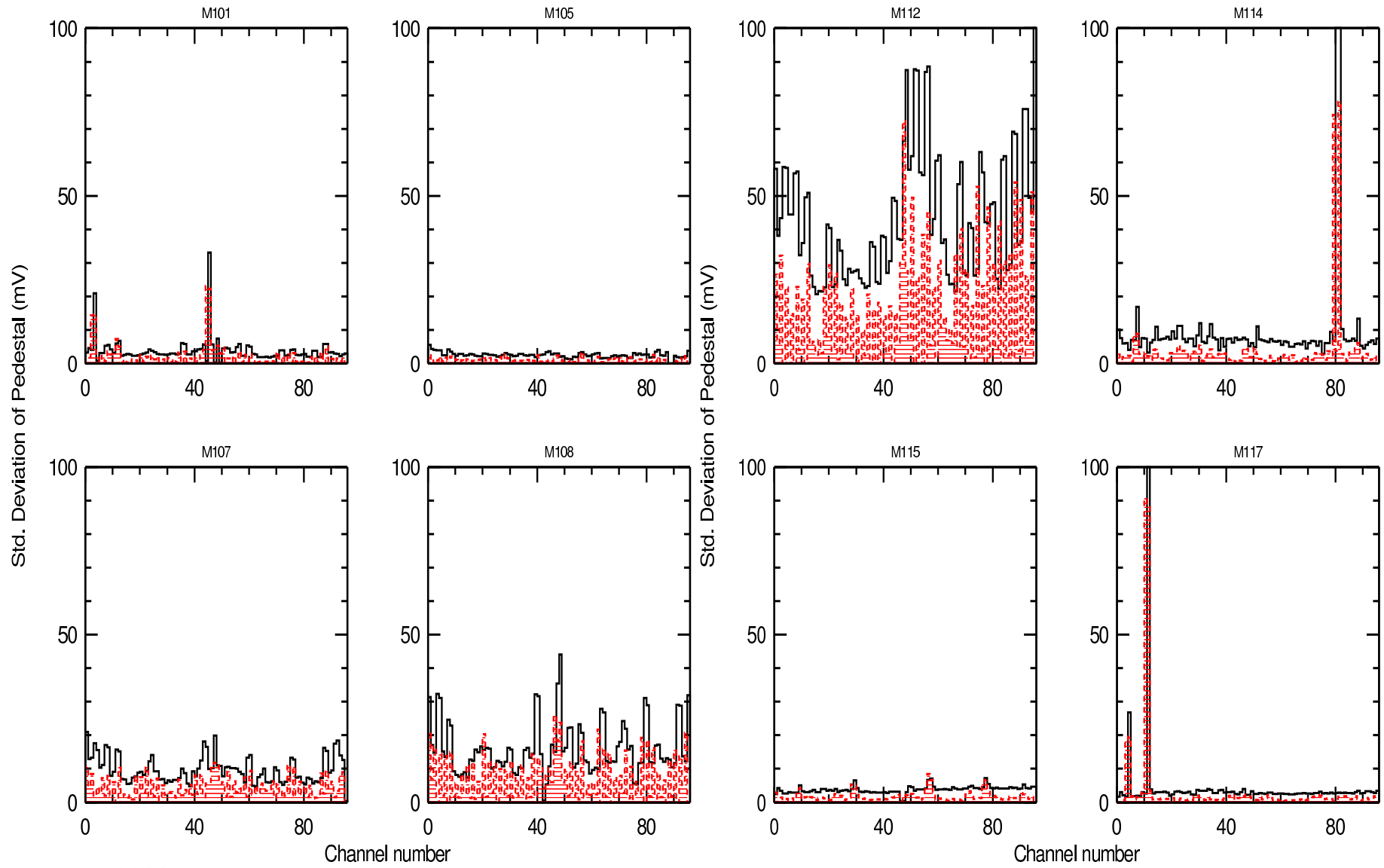
Accessing NuMI Data Offline

- Charlie King's XML-RPC server → BeamData process → minos-acnet.fnal.gov → archive (Bretts talk).
- From archive, Brett converts raw data to a Root TTree.
- Used standalone Root macros to analyze TTree.
- Analyzed BPM, profile monitor and muon/had monitors.

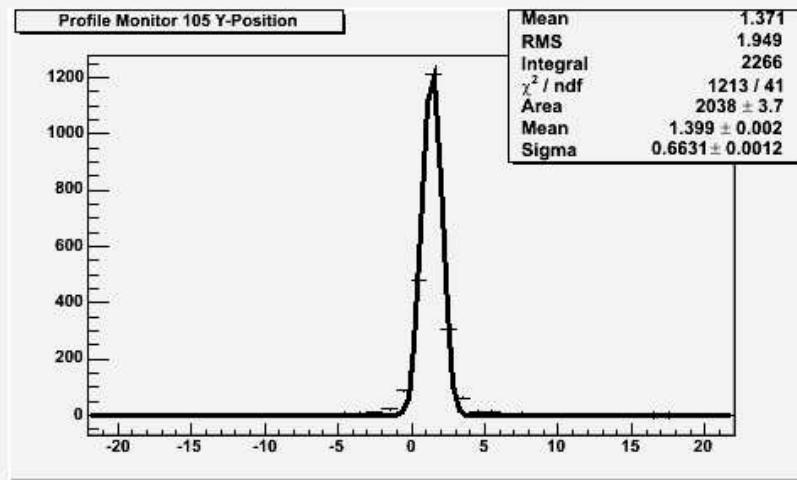
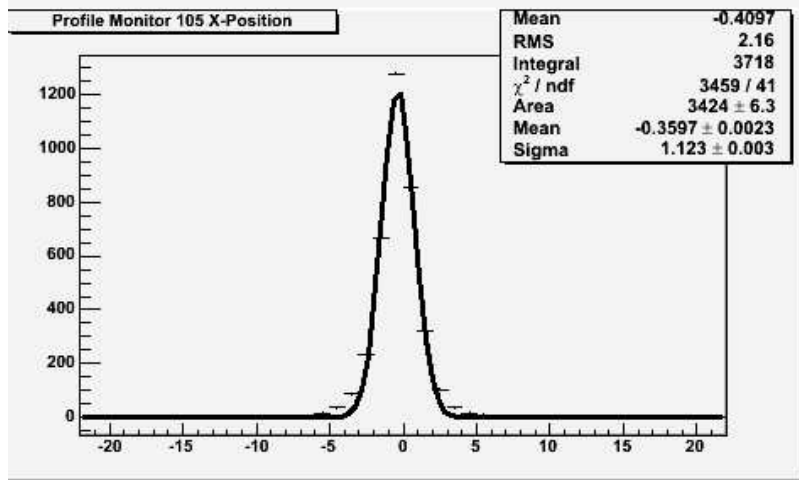
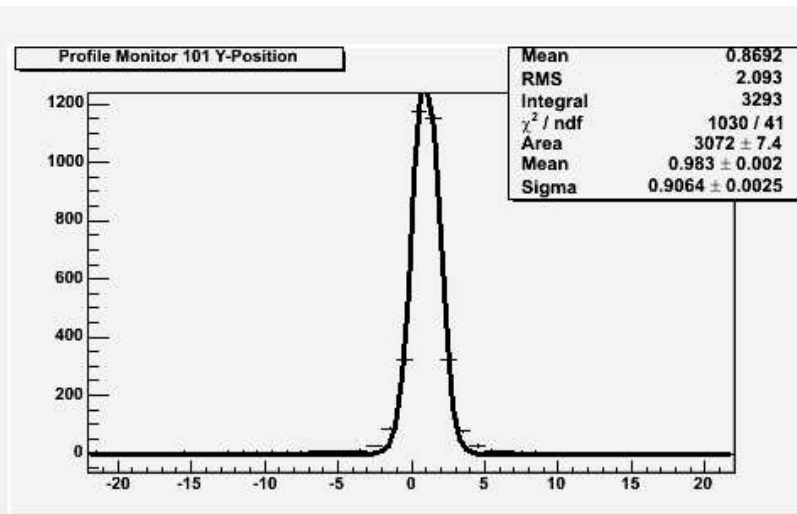
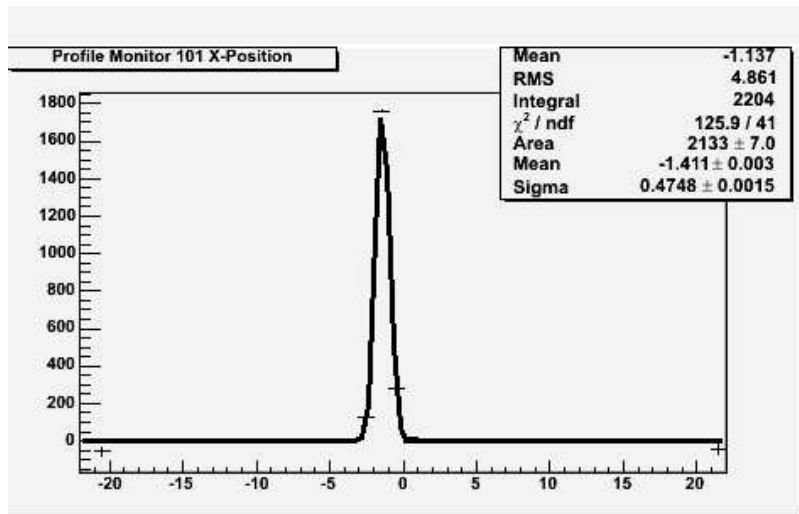
Profile monitor analysis

- **Step 1:** Calculate pedestals from events with v. low beam intensity as measured in TORTGT and/or TOR101.
- **Step 2:** Calculate noise = $\sigma(P_n)$, where P is the pulseheight in channel n .
- **Step 3:** Calculate differential noise = $\sigma(P_n - P_{n+1})$ to identify common mode noise and cross-talk.
- **Step 4:** Loop over events with beam intensity $> 1 \times 10^{11}$ and fit profiles in each spill to a single Gaussian using $\sigma(P_n)$ as the uncertainty on the measurement in each channel.
- **Step 5:** Compare profile widths with beam emittance expectations (P. Lucas).
- **Step 6:** Compare means with nearby BPMs (Sacha, Mark D.)

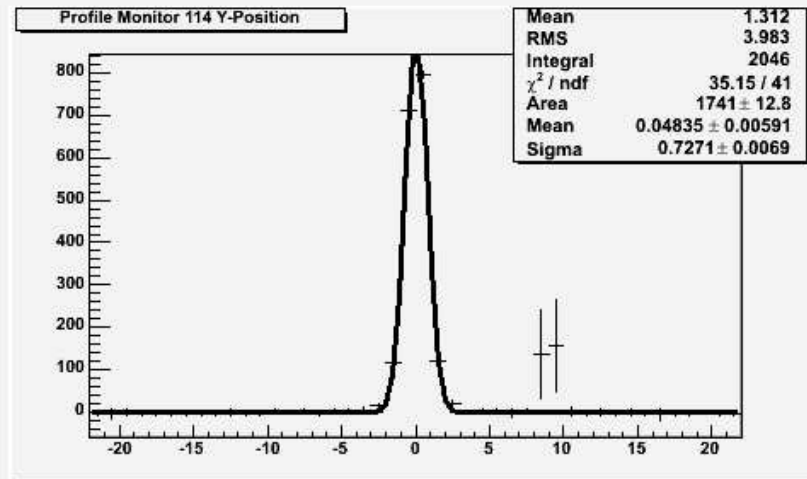
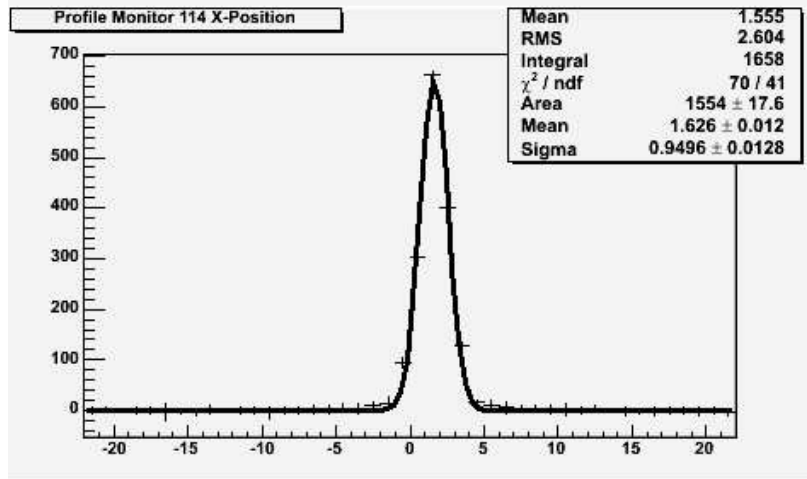
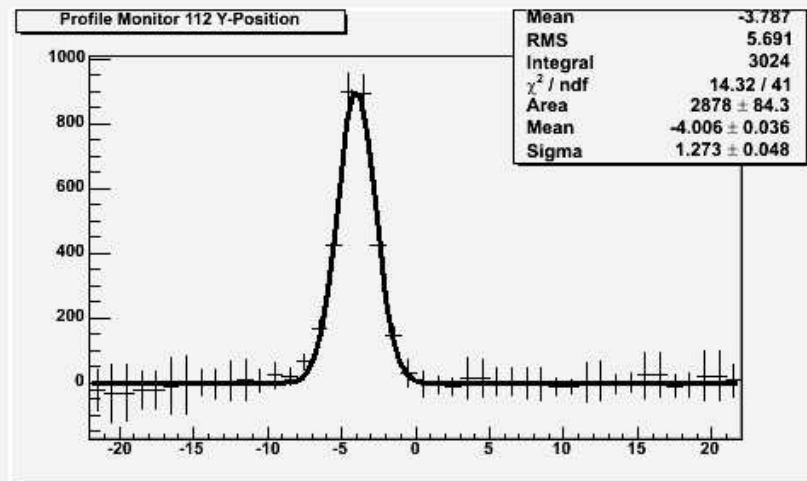
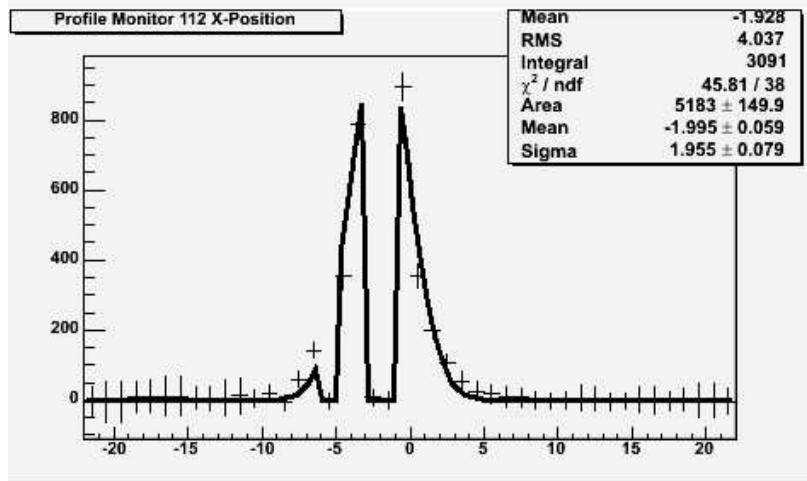
Profile monitor noise/dnoise



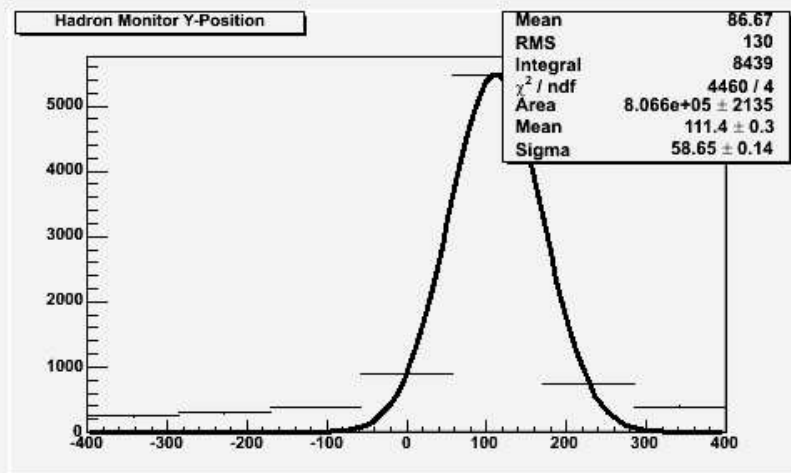
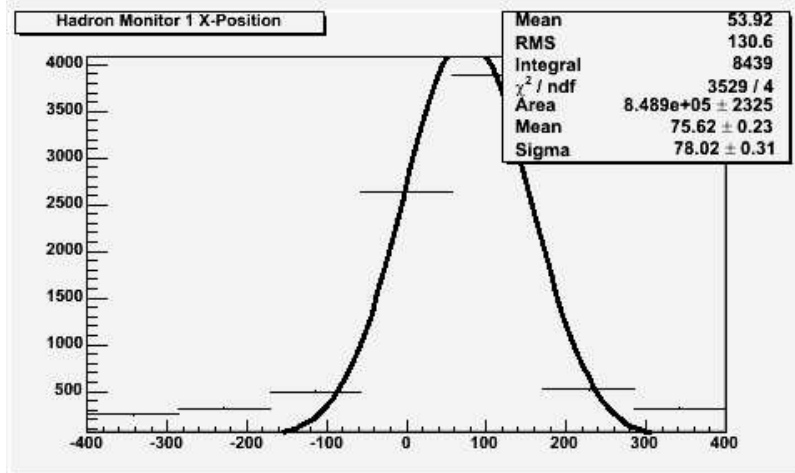
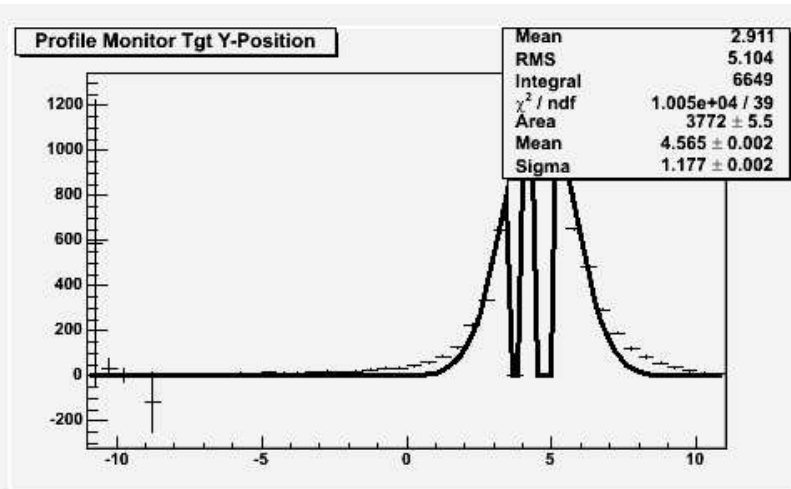
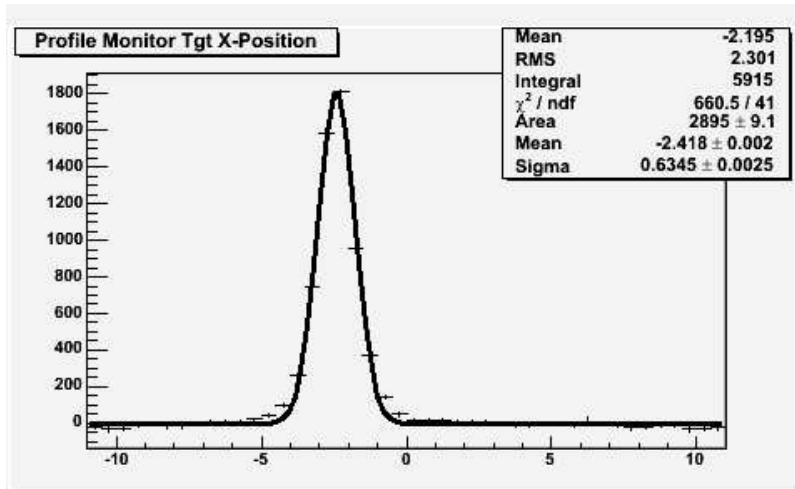
Profile Monitors - Spill 43



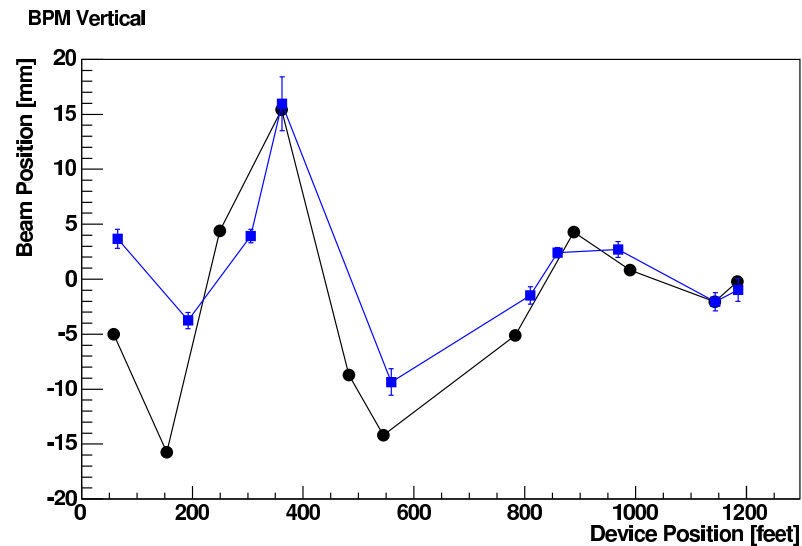
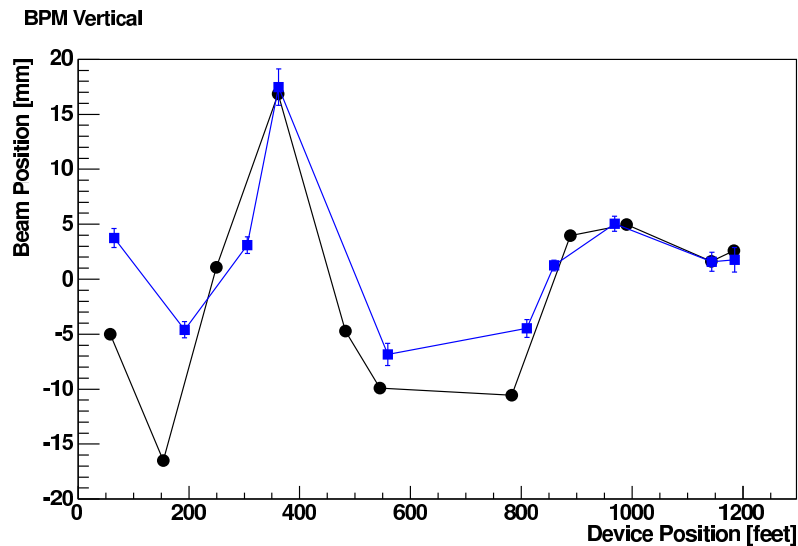
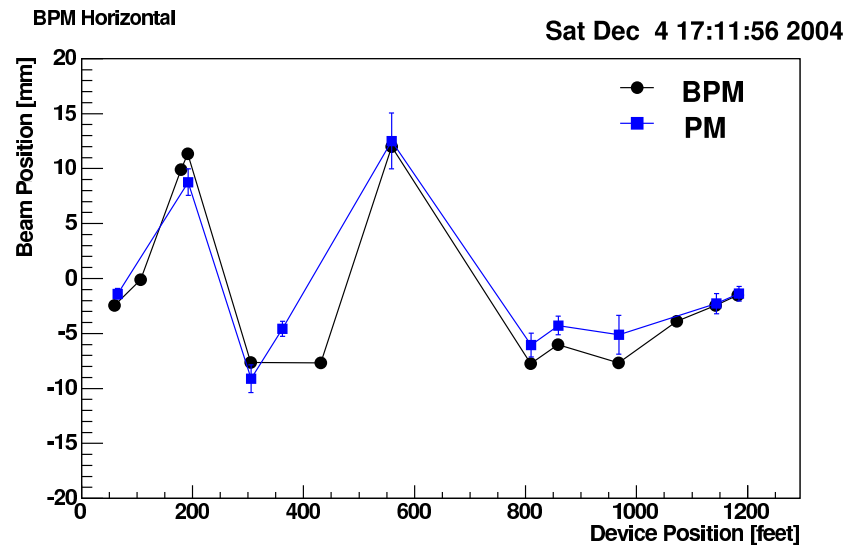
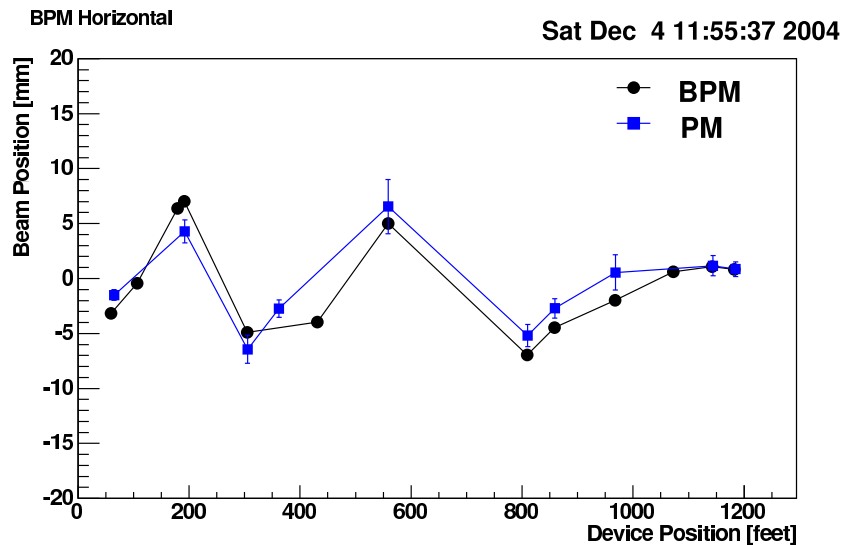
Profile Monitors - Spill 43



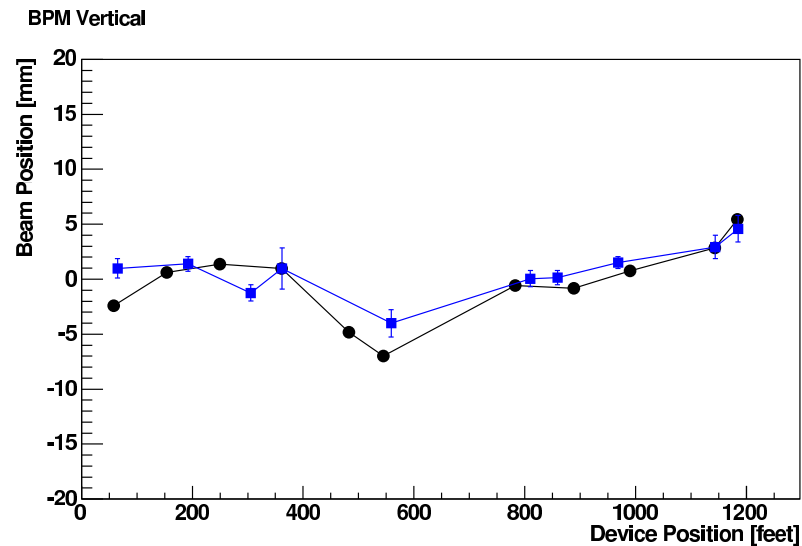
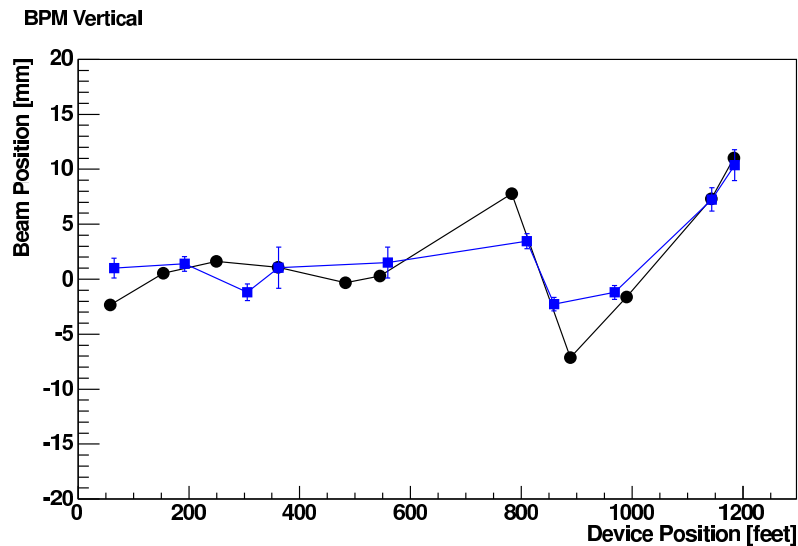
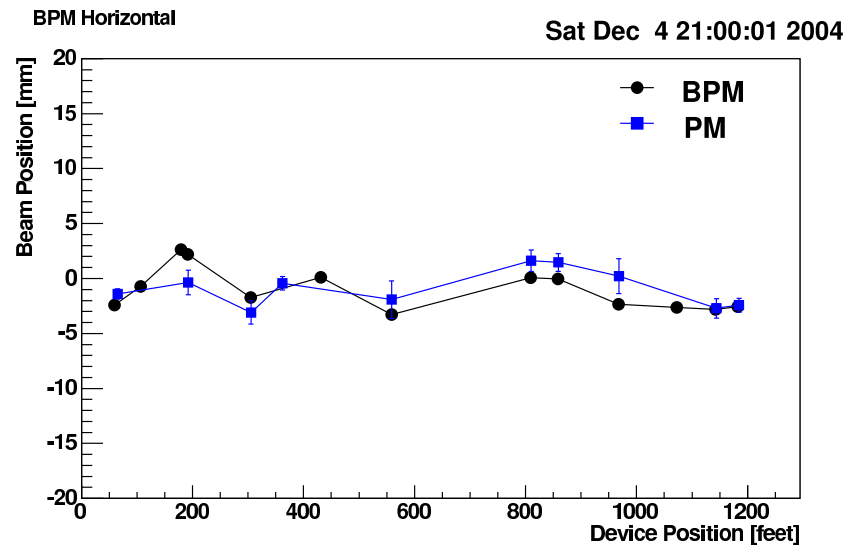
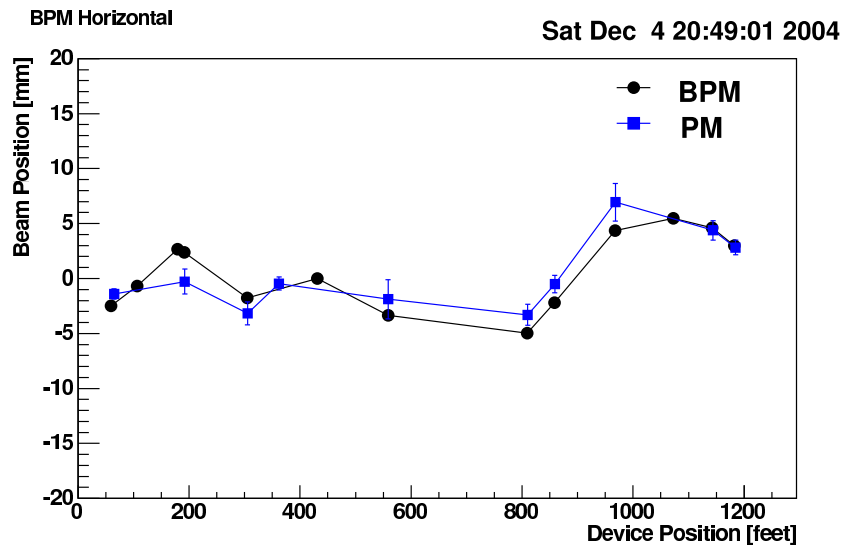
Profile/Had monitor - Spill 43



Profile monitor vs BPMs

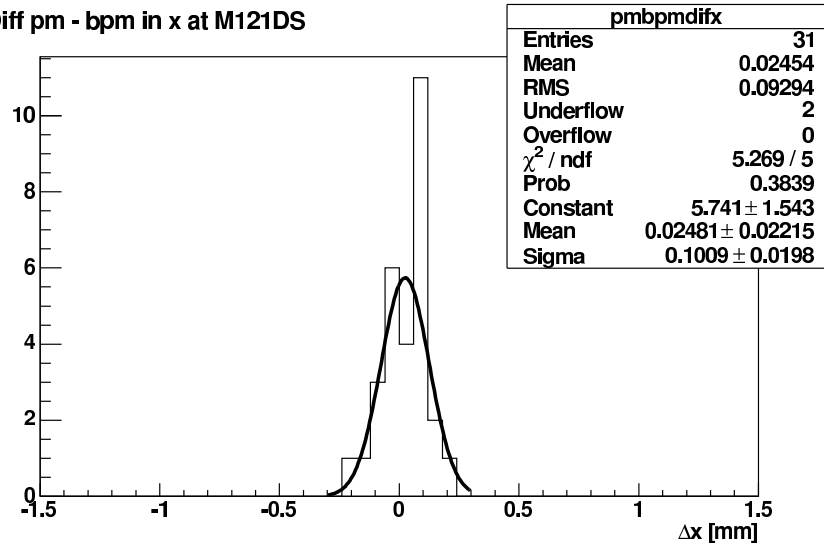


Profile monitor vs BPMs

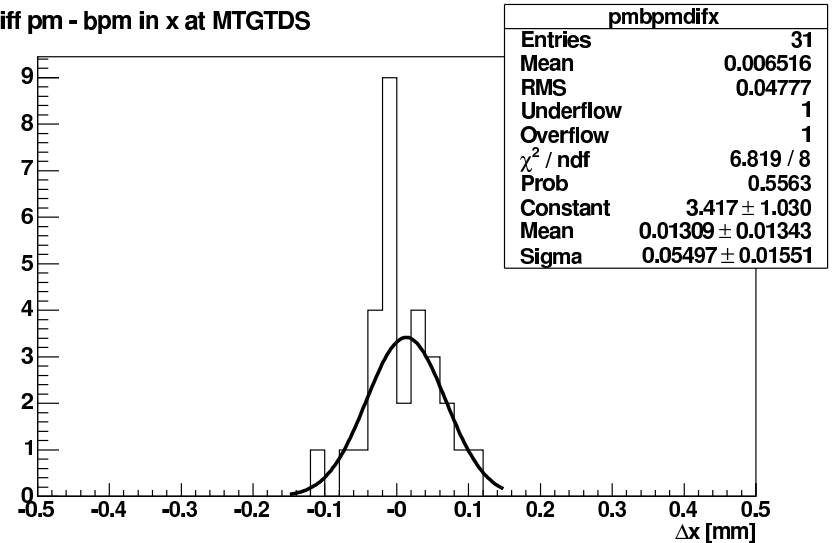


Profile monitor vs BPMs

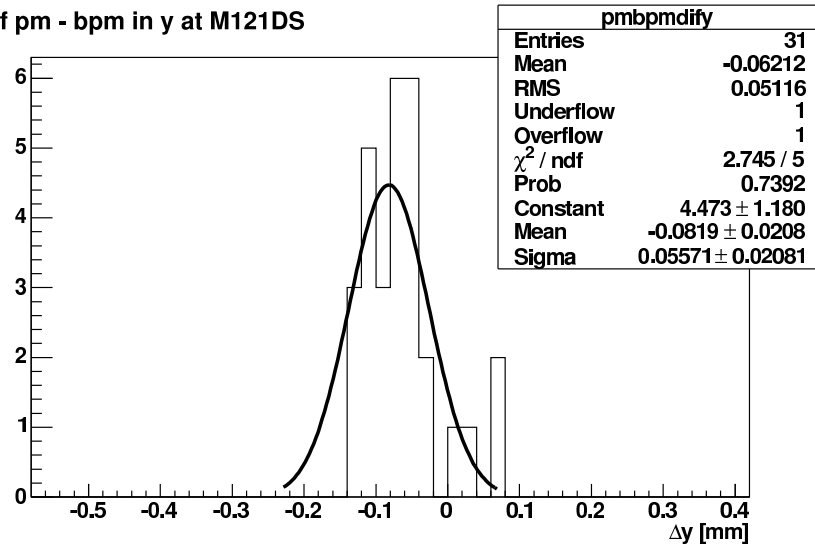
Diff pm - bpm in x at M121DS



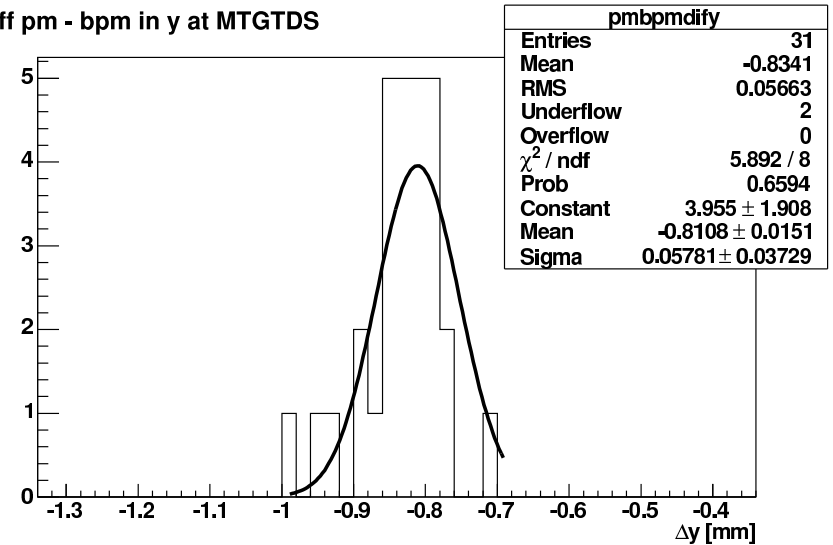
Diff pm - bpm in x at MTGTDS



Diff pm - bpm in y at M121DS

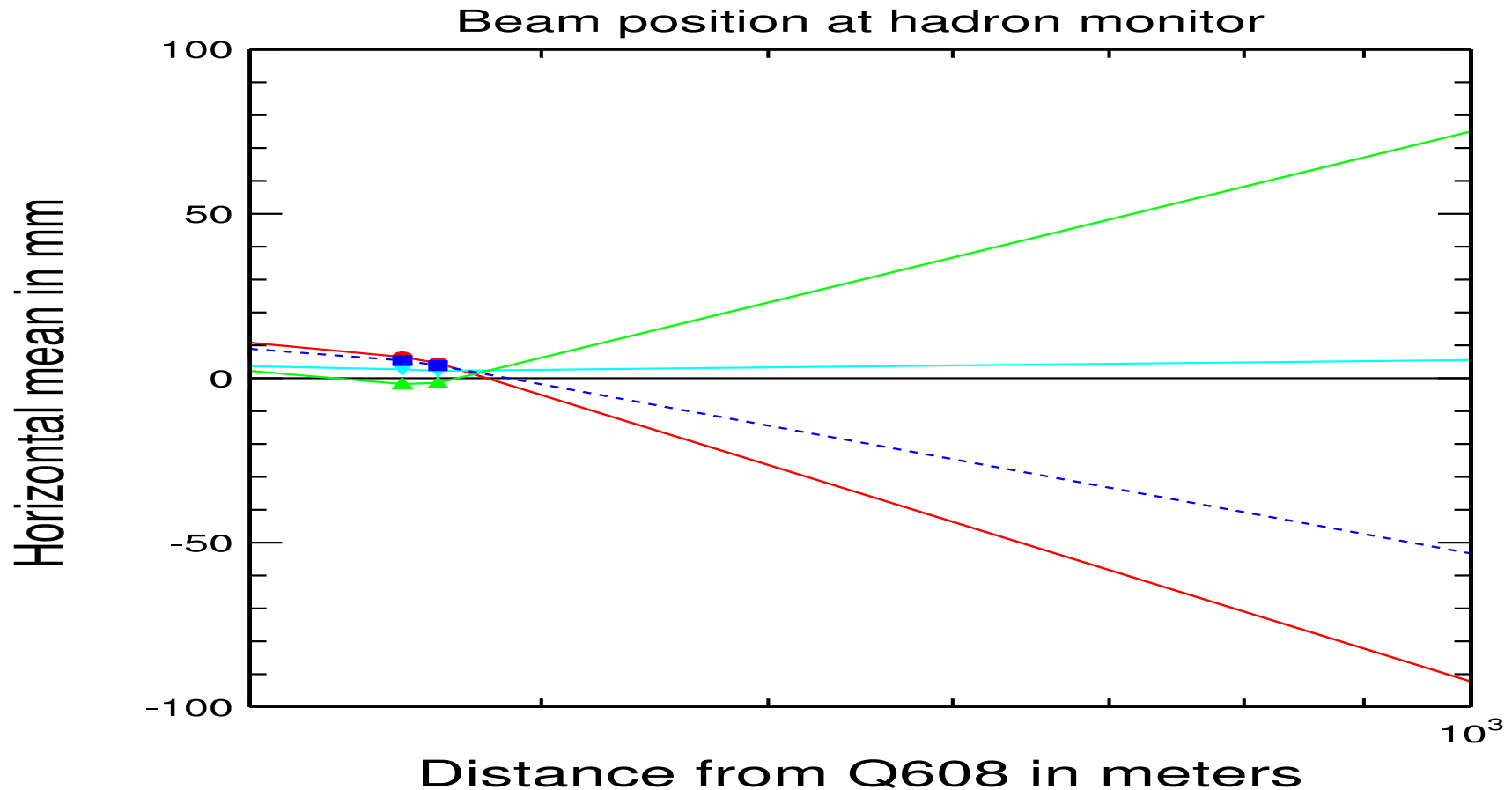


Diff pm - bpm in y at MTGTDS



Hadron Monitor tracks PM

See Sacha's results for a full blown quantitative analysis



Problems: Timing Pathologies

An analysis of the timestamps from the 14 SWICs revealed the following ACNET data timing pathologies in the 385 spills taken:

1. The DAE timestamp of PM114,115,117,121,TGT was 20 seconds ahead of the rest of the SWICS. VME timestamps were identical to ± 30 msec: **1 spill**.
2. The VME timestamps are not in sync on all 14 SWICS. The event contained data from the previous spill on some devices: **10 consecutive spills**. Problem limited to PM114,115,117,121,TGT.
3. No data was returned from PM101,105,107,102,112 and we got an error from the DAE or all zero data: **3 spills**.

NB: Charlie K. has a new service which may have already fixed these problems.

Conclusions

- The JAS3 online NuMI monitor worked well in real conditions. Flexible enough to add new devices and new analysis code and plots on the fly. Some problems with Java memory management/stability. Need to implement a shifter oriented error warning system.
- The ACNET XML-RPC service worked well during data taking. Some timing problems with accessing the data. Its not clear (to me) that the problem is with the server and not further upstream.
- The BeamData process was very easy to run, very reliable and analyzing the NuMI offline data was a blast → if you know some ROOT!.